**MBA 614**

**Merritt Aho**

**Final Project: Personal Budget Worksheet**

Description:

This spreadsheet accesses the user’s credit card website, downloading balance information and all transactions ytd. The program then processes that transaction information and prompts the user to classify each transaction according to the user’s budget categories. The program generates and displays current spending amounts for each budget category and also ytd spending under each budget category.

Quick Stats:

475 lines of code

19 sub procedures

2 web queries

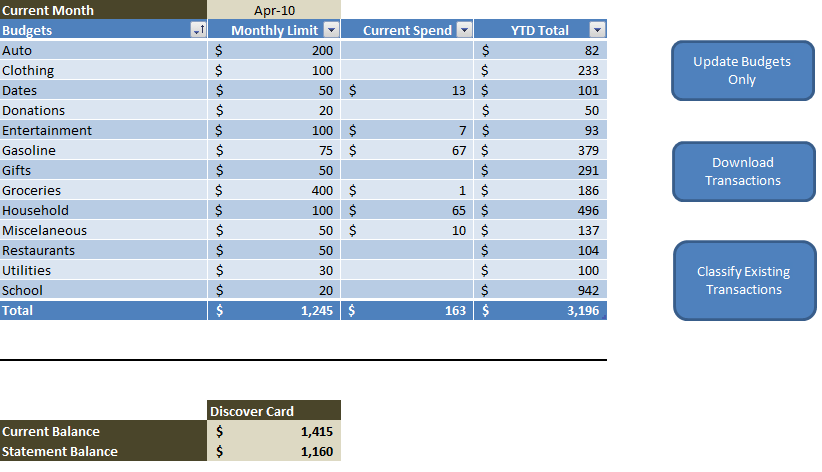
≈24 hours of work

Limitations:

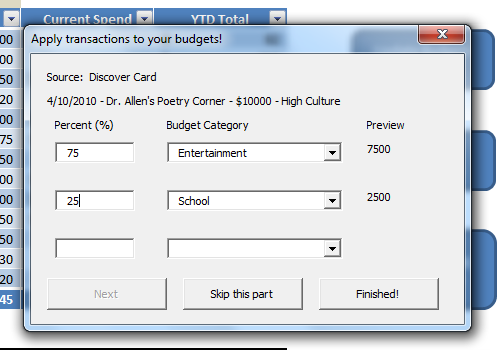
This spreadsheet will not work off of my hard drive. The first reason for this is that I have removed my user name and password information that the program uses to access my financial data. The second reason is that the web query that imports the financial data into the spreadsheet is dependent upon a specific file path on my machine.

Detailed Explanation:

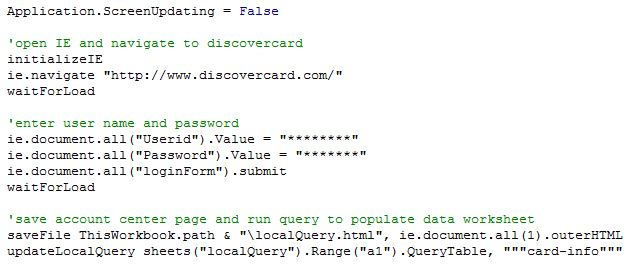
1User Home Screen. Update Budgets button simply re-analyzes currently coded data. Download transactions button gets new data and prompts the user to code it. Classify Transactions button prompts the user to finish coding previously dl data.



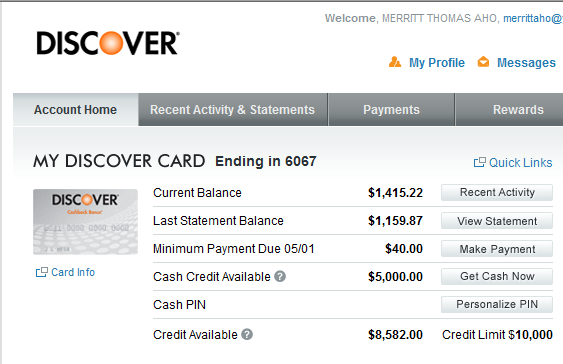
This is the user form. It is very dynamic. The form will initialize after new transaction data has downloaded. However, the user may choose to classify their transactions to budgets at a later time. In that case, the user 'Skips' and may return later.



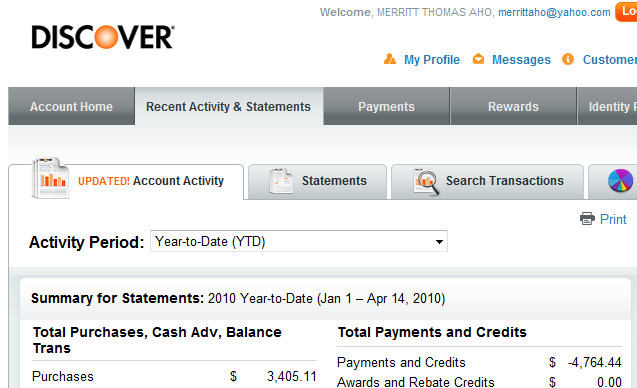
This code initializes IE and logs onto the website. It then saves the page immediately to an html file on the user's hard drive.



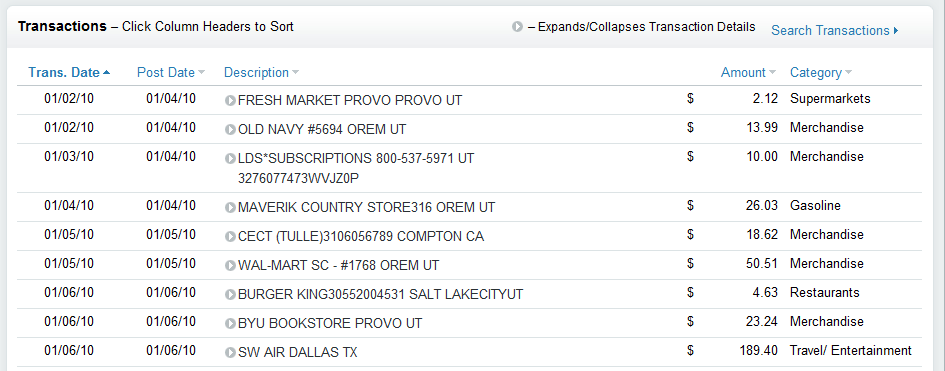
This is the first screen that IE 'sees' when it logs onto my credit card company's website. This page is saved to disk and the balance information is uploaded to the spreadsheet via a web query on the html file.



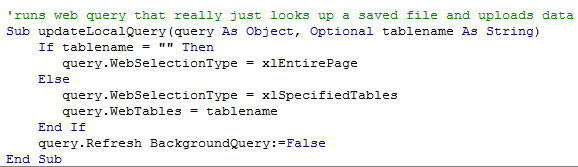
This is the second screen that IE 'sees' after logging in. It contains all of the credit card transaction data. This page is saved to disk and then one of the tables is uploaded to the spreadsheet via a web query on the html file.



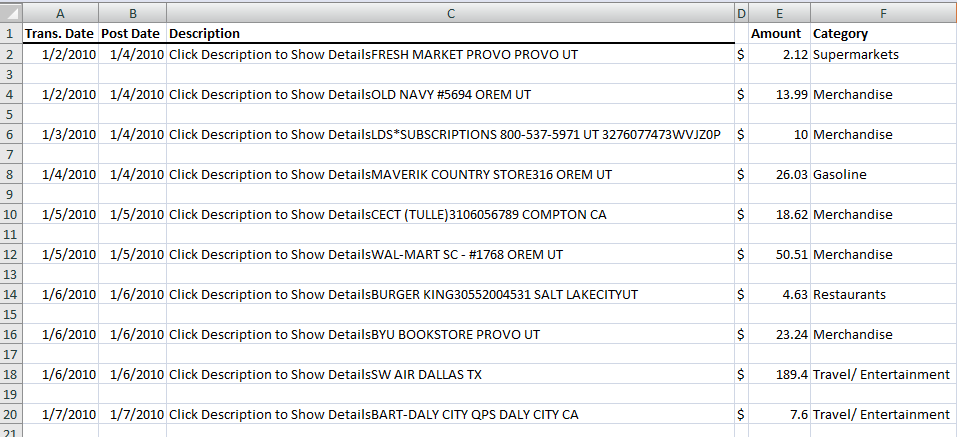
Transaction data table as it appears in IE at the credit card website.



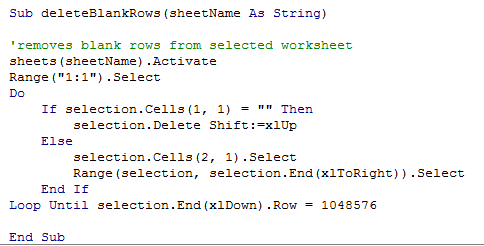
This code updates the query inside the spreadsheet by pulling either a table, or an entire page from an html file saved on the user's hard drive.



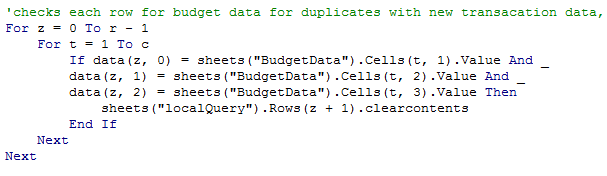
This is the raw data when it has been imported by the web query.



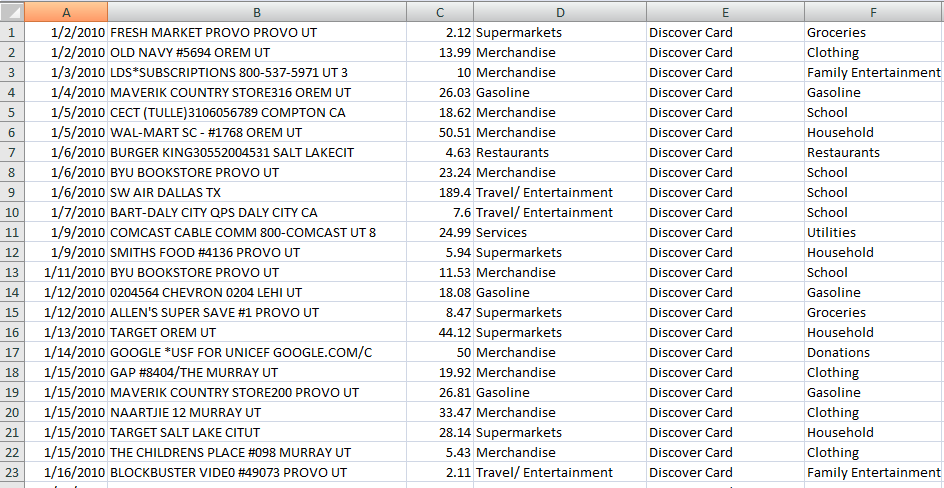
Sub procedure deletes empty rows from a specified worksheet.



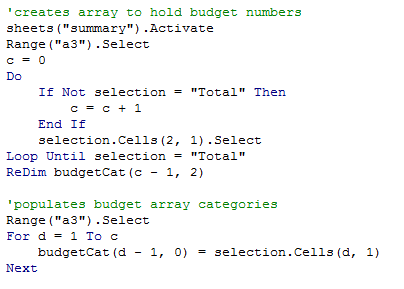
A crucial piece of code that removes duplicates from web query data before it is combined with existing transaction data.



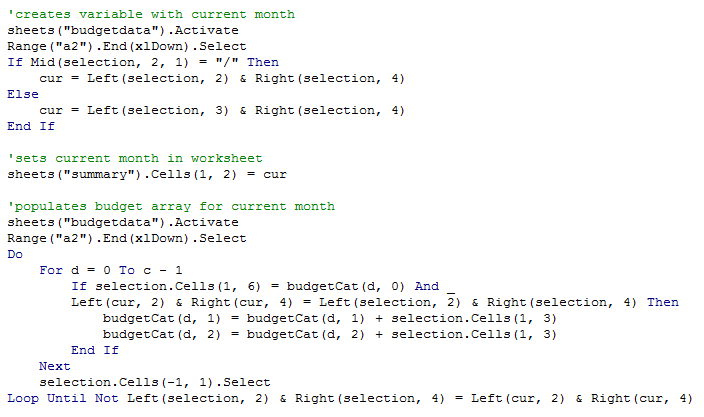
This is how the transaction data looks once it has been scrubbed for duplicates and unuseful information. It is added to other historical transactions and sorted by date.



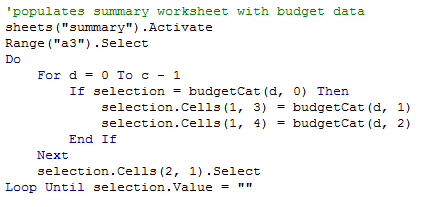
The code uses arrays to hold and compile data for each of the budget categories.



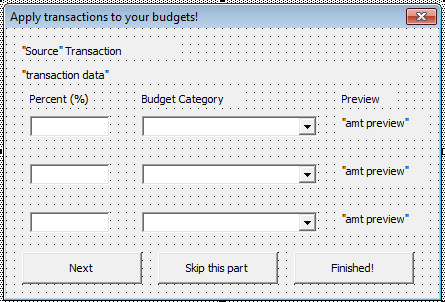
One of the greatest challenges in writing this code was dealing with the transaction dates. This code manages to read only the month and year of the most recent transactions and then adds current transactions to budget expenses by category.



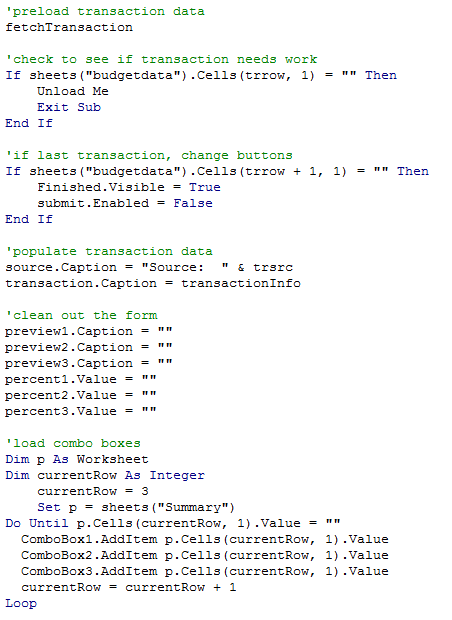
Code populates the main user page with numbers.



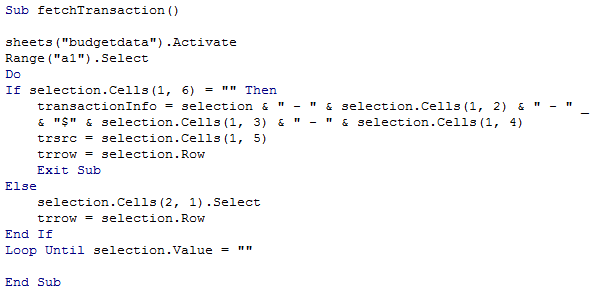
Set-up for the user form.



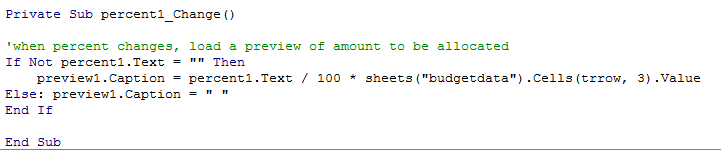
Initialize User Form. This code calls a sub which fetches an uncoded transaction and populates the form with it.



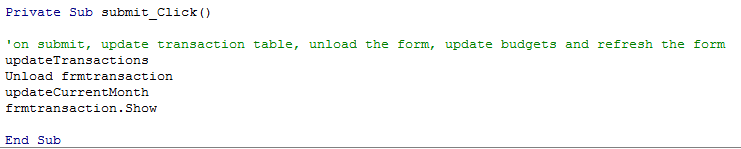
The code that fetches an uncoded transaction.



Whenever a user wants to split a transaction between budgets, the form automatically creates a preview of the transaction using this code.



When a user finishes coding a transaction they push 'Next' (unless it is the final transaction, in which case 'Finish' is the only option. This code calls a sub to process the coding and refreshes the form with new data.



This code updates the data table with the new coded information. If the transaction was split among different categories, that is indicated in another column.

